



# جامعة طرابلس كلية تقنية المعلومات



مقدمة في قواعد البيانات

**Introduction to Databases**

**ITGS228**

**[h.ebrahem@uot.edu.ly](mailto:h.ebrahem@uot.edu.ly)**

الأستاذ - حسن علي حسن

المحاضرة الخامسة عشر – لغة الاستعلام المهيكلية

**Structured Query Language SQL**

# مواضيع المحاضرة

- لغة الاستعلام المهيكله SQL Structured Query Language
- انشاء واستخدام وحذف قاعدة بيانات باستخدام MySQL
- انشاء جدول
- جملة select
- دمج البيانات بين استعلامين
- استخدام الربط join.

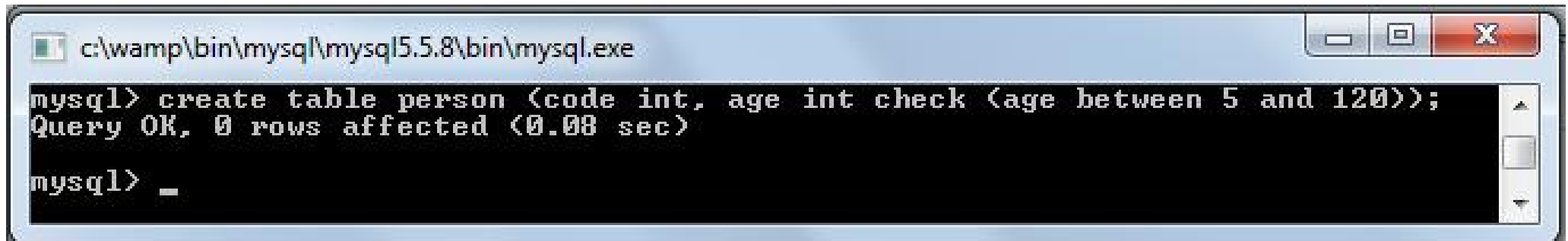
# Introduction to SQL

**Structured Query Language**

لغة الاستعلام المهيكل

# لغة الاستعلام المهيكله SQL Structured Query Language

- تعرف لغة الاستعلام المهيكله SQL Structured Query Language بأنها لغة التعامل مع قواعد البيانات وتعتمد عليها كافة التطبيقات التي تتعامل مع قواعد البيانات العلائقية.
- تم تطوير لغة SQL انطلاقاً من النموذج العلائقي لـ كودد (CODD) المعتمد على الجبر العلائقي.
- SQL تمكنك هذه اللغة من إدارة قواعد البيانات بشكل كامل وإجراء جميع العمليات القياسية كإنشاء الجداول وتعبئتها بالبيانات، أو إجراء الاستعلامات عليها وكذلك الربط بين الجداول المختلفة.
- توجد بعض الارشادات عند كتابة جمل SQL :
  1. يمكن كتابة الجمل بالاحرف الصغيرة و الكبيرة.
  2. يفصل بين اسماء الحقول بالفاصلة (,).
  3. الاعلان عند نهاية الجملة بالفاصلة المنقوطة (;).

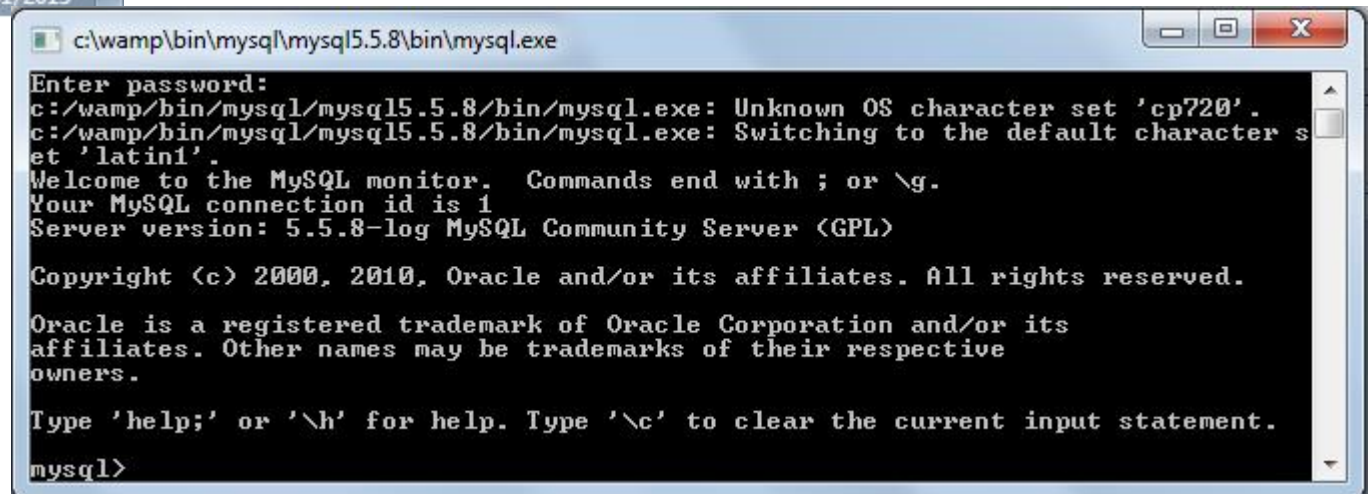
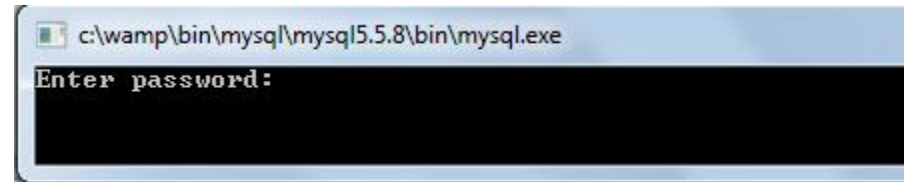
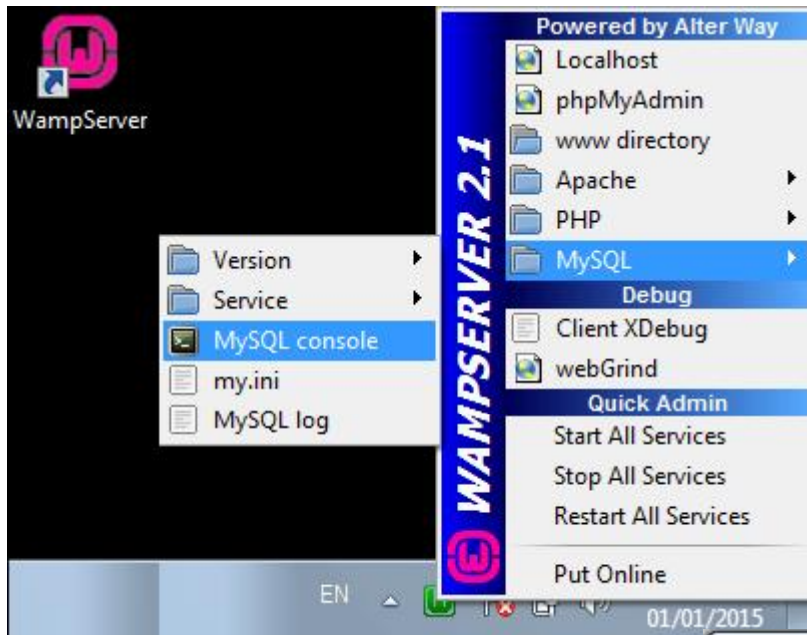


```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
mysql> create table person (code int, age int check (age between 5 and 120));
Query OK, 0 rows affected (0.08 sec)
mysql> _
```

# MySQL إنشاء قاعدة بيانات

➤ في البداية لتطبيق اوامر SQL سنستخدم برنامج WampServer ، سأقوم بعملية توضيح

كيف يتم تشغيله على الجهاز كالآتي:



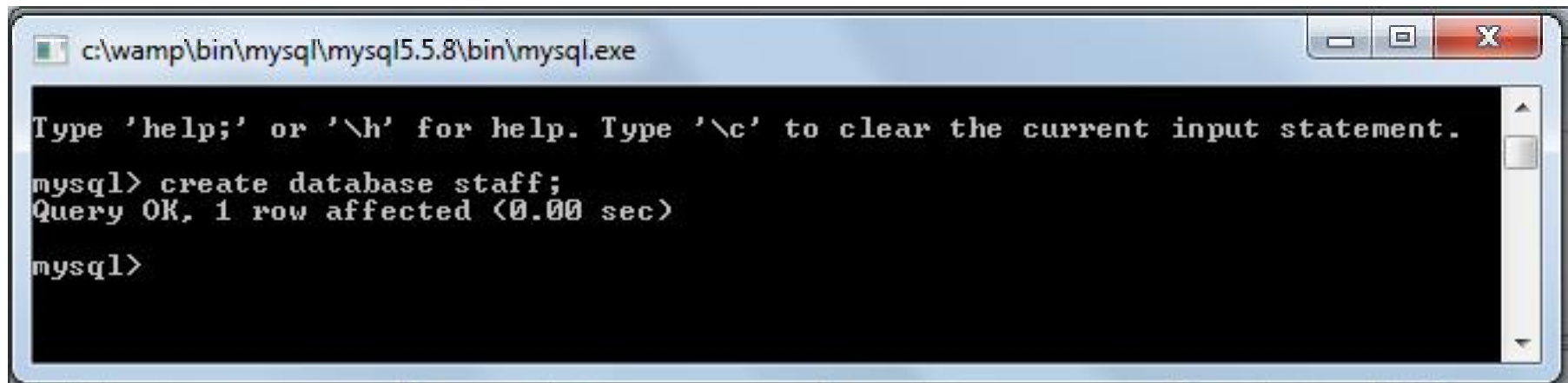
# MySQL انشاء قاعدة بيانات

➤ في بداية استخدام SQL يتم انشاء قاعدة البيانات باستخدام امر

➤ CREATE DATABASE

**CREATE DATABASE db\_name;**

• الامر السابق يقوم بانشاء قاعدة بيانات باسم معين **db\_name**.



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database staff;
Query OK, 1 row affected (0.00 sec)

mysql>
```

• في نظام ادارة قواعد البيانات Oracle: ان استخدام هذا الامر مع اسم قاعدة بيانات موجودة مسبقا يؤدي الى إلغاء و إستبدال تلك القاعدة.

# MySQL حذف قاعدة بيانات

## ➤ DROP DATABASE

**DROP DATABASE db\_name;**

- الامر السابق يقوم بحذف قاعدة البيانات db\_name وكل ما تحتويه من جداول.
- Be VERY careful with this command!



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
mysql> drop database staff;
Query OK, 0 rows affected (0.00 sec)
mysql>
```

# MySQL استخدام قاعدة بيانات

➤ عندما تريد العمل على قاعدة بيانات معينة في البداية يتم تحديدها باستخدام امر

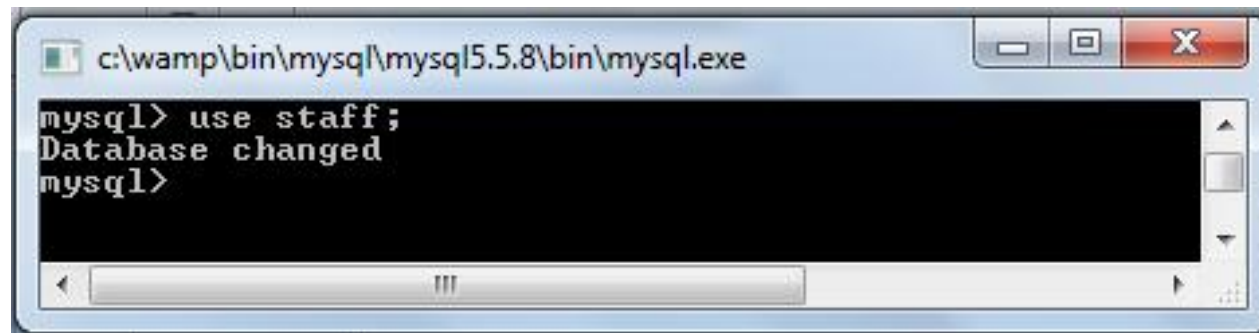
➤ USE DATABASE

**USE db\_name;**

الامر السابق يخبر MySQL بان قاعدة البيانات db\_name قاعدة بيانات رئيسية.  
استخدام الامر السابق لا يمنع من أن نصل الى جداول في قواعد بيانات أخرى.  
المثال التالي يبين كيفية الوصول من قاعدة بيانات الى أخرى

```
mysql> USE db1;
```

```
mysql> SELECT author_name,editor_name FROM author,db2.editor  
WHERE author.editor_id = db2.editor_id;
```





# MySQL إنشاء جدول

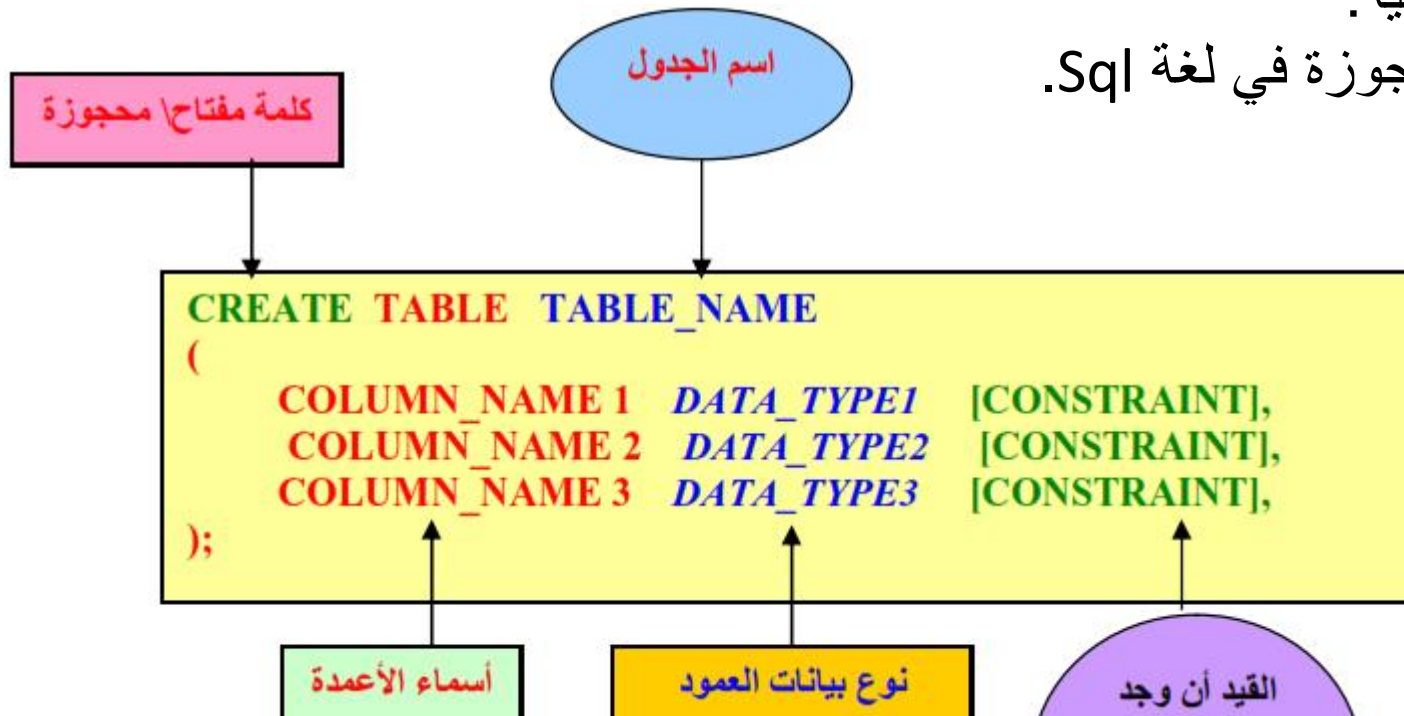
➤ بعد ما إنشاء قاعدة بيانات وقمنا باستدعائها للاستخدام، نقوم بإنشاء جداول بداخلها.

➤ من قواعد تسمية الجداول الآتية:

1. أن لا يكون الاسم كلمة محجوزة في لغة Sql.

2. أن لا يبدأ الجدول برقم.

الصيغة العامة هي



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe  
mysql> use staff;  
Database changed  
mysql> create table TestOne (id int, fullname varchar(50), birthdate datetime)  
-> ;  
Query OK, 0 rows affected (0.03 sec)  
mysql> _
```

# MySQL إنشاء جدول

➤ **انواع البيانات Data Type :** كل حقل من الحقول في الجداول له نوع معين من البيانات، حسب القيمة المراد تخزينها. نبين البعض منها:

➤ **الانواع الرقمية : تستخدم لتخزين القيم الرقمية**

1. Int : يستعمل مع الارقام الصحيحة ويأخذ 4 بايت.
2. Smallint : يستعمل مع الارقام الصحيحة ويأخذ 2 بايت.
3. Real : يستعمل مع الارقام العشرية ويأخذ 4 بايت.
4. float : يستعمل مع الارقام العشرية ويأخذ 8 بايت.

➤ **الانواع النصية : تستخدم لتخزين النصوص**

➤ **Varchar(n) :** يستخدم لتخزين عدد كبير من البيانات ويتميز بتغيير حجمه آليا على عدد البيانات التي يحتويها.

➤ **Char(n) :** يستخدم لتخزين النصوص حسب القيمة التي بين القوسين.

➤ **التاريخ والوقت : تستخدم لتخزين الوقت والتاريخ**

➤ **Datetime :** يستخدم لتخزين التواريخ والاوقات ويأخذ 8 بايت.

# جملة select

## الشكل العام للجملة

**SELECT** field\_1,field\_2,... **FROM** table\_name **WHERE** condition

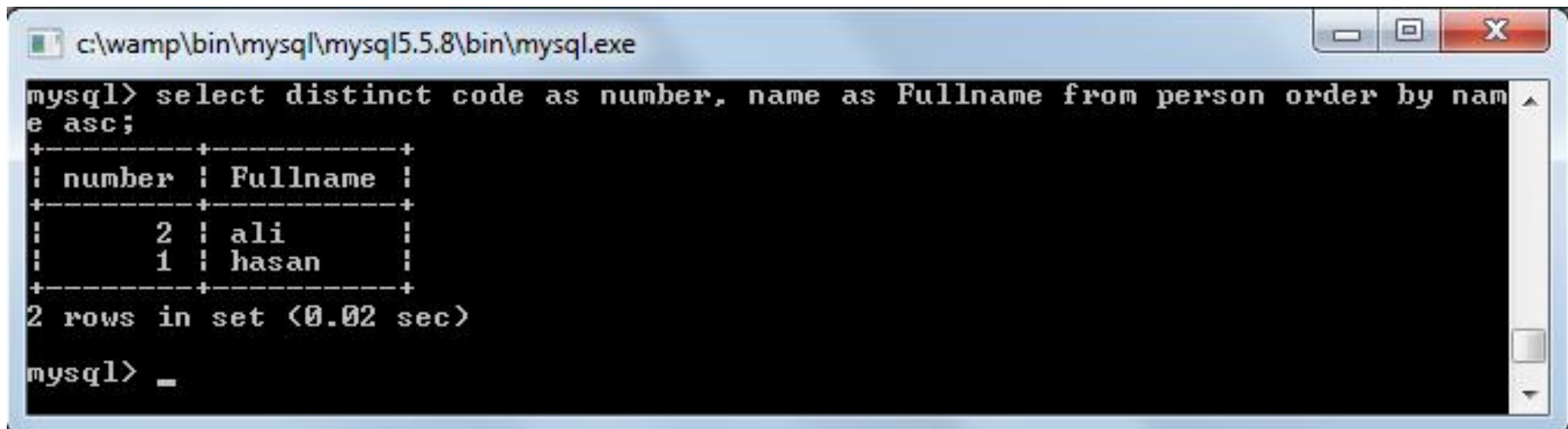
- تعتبر تعليمة **SELECT** من أشهر تعليمات اللغة وأكثرها استخداماً. تستخدم هذه التعليمة لاستعادة وانتقاء مجموعة من البيانات من قاعدة البيانات وذلك بإعادة جدول يحتوي البيانات المطلوبة.
- جملة الاستفسار **SELECT** تحدد أسماء الحقول أو الأعمدة التي سوف تظهر في النتيجة.
- عبارة **FROM** هي عبارة ضرب كارتيزي في حالة أكثر من جدول حيث تحدد أسماء الجداول التي سوف يتم استخراج المعلومات منها.
- عبارة **WHERE** فهي إختيارية وتحدد معايير الاختيار من الجداول المذكورة في جملة **FROM** .
- جملة **SELECT** تستخدم للإسقاط **Projection** أما جملة **WHERE** فتعبر عن الاختيار **Restriction**.

# جملة select

## الشكل العام للجملة

**SELECT** field\_1,field\_2,... **FROM** table\_name **WHERE** condition

- تستخدم إشارة \* كبديل لأسماء الحقول. أي لعرض كل الحقول بالجدول.
- يستخدم تعبير DISTINCT لاستعادة جميع السجلات مع إلغاء التكرار في السجلات المعادة.
- يستخدم التعبير ORDER BY لترتيب السجلات المعادة ترتيباً تصاعدياً أو تنازلياً حسب التعبير المرافق المستخدم: ASC للترتيب التصاعدي أو DESC للترتيب التنازلي.
- في حال الرغبة باستخدام أسماء بديلة لحقول جدول القيم المعادة نستخدم التعبير AS.



```
c:\wamp\bin\mysql\mysql5.5.8\bin\mysql.exe
mysql> select distinct code as number, name as Fullname from person order by name asc;
+-----+-----+
| number | Fullname |
+-----+-----+
|      2 | ali      |
|      1 | hasan    |
+-----+-----+
2 rows in set (0.02 sec)

mysql> _
```

# جملة select

تأتي مع جملة select كلمة where لإستعادة مجموعة من السجلات حسب الشرط.  
**SELECT** field\_1,field\_2,... **FROM** table\_name **WHERE** condition

- يمكن ان يتضمن الشرط عمليات مقارنة < , > , <= , >= , <> , OR , AND , NOT
- تستخدم كلمة like ضمن العبارة الشرطية، غالبا ما تستخدم هذه الكلمة مع اشارة % والتي تضاف الى القيمة كبديل عن أي رقم أو حرف.
- تستخدم كلمة between ضمن العبارة الشرطية، كشرط لقيمة محصورة بين قيمتين محدودتين.

```
mysql> select name from person where name like '%n%';
+-----+
| name |
+-----+
| hasan |
| Hane |
| Anwer |
+-----+
3 rows in set (0.00 sec)

mysql>
```

```
5 rows in set (0.00 sec)

mysql> select * from person where code between 3 and 4;
+-----+-----+
| code | name |
+-----+-----+
| 3    | Akram |
| 4    | Hane |
+-----+-----+
2 rows in set (0.02 sec)

mysql>
```

```
mysql> select * from person where name like 'h%' and code between 2 and 4;
+-----+-----+
| code | name |
+-----+-----+
| 4    | Hane |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

# مثال

مثال: المطلوب عرض اسماء جميع المستخدمين اللذين استعاروا الكتاب رقم 10

**SELECT** uname **FROM** user u, borrow b **WHERE** u.uid=b.uid **AND** b.bid = 10

user

UID	UNAME	RATING	AGE
22	Haitham	7	45
31	Khaled	8	55.5
58	Zaid	10	35

borrow

UID	BID	borrow_date
22	10	2008-05-20
31	20	2008-02-12

لمزيد من فهم جملة الاستفسار يمكن ان نتصور ان تنفيذ الجملة داخل الحاسب سيتم كما يلي:

1. يتم تنفيذ الضرب الكارتيبي بين جميع الجداول المذكورة في جملة **FROM**.
2. يتم الغاء جميع الاسطر التي لا تحقق الشروط المذكورة في جملة **WHERE**.
3. يتم الغاء جميع الاعمدة التي لا تظهر في جملة **SELECT**.

- هنا قمنا بتعريف اسمان مستعاران الاول u و هو اسم مستعار لجدول user و الثاني b كاسم مستعار لجدول borrow، قمنا باستخدام الاسم المستعار بعد جملة where مع اسم الحقل.

## مثال

**SELECT** uname **FROM** user u, borrow b **WHERE** u.uid=b.uid **AND** b.bid = 10

**user**

UID	UNAME	RATING	AGE
22	Haitham	7	45
31	Khaled	8	55.5
58	Zaid	10	35

**borrow**

UID	BID	borrow_date
22	10	2008-05-20
31	20	2008-02-12

• إذا أردنا تطبيق الخطوات المذكورة السابقة نبدأ بعملية الضرب الكارتيزي والتي تنتج الجدول:

b.UID	BID	DATE	u.UID	UNAME	RATING	AGE
22	10	2008-05-20	22	Haitham	7	45
31	20	2008-02-12	22	Haitham	7	45
22	10	2008-05-20	31	Khaled	8	55.5
31	20	2008-02-12	31	Khaled	8	55.5
22	10	2008-05-20	58	Zaid	10	35
31	20	2008-02-12	58	Zaid	10	35

• ثم نحذف جميع الأسطر التي لا تحقق الشروط في جملة **WHERE** فتصبح النتيجة كما يلي:

b.UID	BID	DATE	u.UID	UNAME	RATING	AGE
22	10	2008-05-20	22	Haitham	7	45

• و من ثم يتم حذف جميع الأعمدة عدا العمود المذكور في جملة **SELECT** لتصبح النتيجة:

UNAME
Haitham

# دمج البيانات بين استعلامين

العمليات على المجموعات (الاتحاد UNION ، التقاطع INTERSECTION والفرق EXCEPT) تستخدم العمليات UNION ، INTERSECTION ، EXCEPT في دمج البيانات بين استعلامين، وتختلف النتائج بحسب التعبير المستخدم. الشكل العام:

```
select columnA,columnB from tableA  
Operator  
Select columnC,columnD from tableB;
```

التعبير UNION:

يستخدم التعبير Union لدمج البيانات التي يُعيدها استعلامان، حيث يُستخدم وفق الصيغة:

```
select columnA,columnB from tableA  
UNION  
Select columnC,columnD from tableB;
```

في هذه الحالة سيتم إدراج السجلات المعادة من الاستعلام الأول مع السجلات المعادة من الاستعلام الثاني، مع إهمال السجلات المكررة.



## دمج البيانات بين استعلامين- الاتحاد UNION

مثال: نفرض لدينا جدول للموظفين به ارقام واسماء ودرجات الاختبار وجدول للمدراء به اسماء وارقام و درجات الاختبار. المطلوب عرض اسماء ودرجات الموظفين المتحصلين على درجات اكثر من 50 وأسماء ودرجات المدراء المتحصلين على درجات اكثر من 60 في جدول واحد.

```
mysql> select * from manager;
+-----+-----+-----+
| magid | magname | maggrade |
+-----+-----+-----+
| 100   | akram   | 55       |
| 200   | hane    | 85       |
| 200   | lugin   | 90       |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from employ;
+-----+-----+-----+
| empid | empname | empgrade |
+-----+-----+-----+
| 10    | hasan   | 80       |
| 20    | ali     | 70       |
| 30    | adam    | 40       |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select empname as name, empgrade as grade from employ where empgrade > 50
union select magname, maggrade from manager where maggrade > 60;
```

```
+-----+-----+
| name  | grade |
+-----+-----+
| hasan | 80     |
| ali   | 70     |
| hane  | 85     |
| lugin | 90     |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

# دمج البيانات بين استعلامين- الاتحاد UNION

مثال: المطلوب عرض أسماء الطلاب اللذين درسوا المادة " calculus 1 " أو " calculus 2 ".  
نفرض أن لدينا ثلاث جداول في قاعدة البيانات التالية:

جدول الطالب Student

sid	sname	age	avg-marks	level
21	Ahmad	20	66	3
27	Mohammed	23	78	4
29	Tawfiq	23	72	4
34	Ammar	19	78	1
36	Mahmoud	20	76	2
46	Ruba	22	79	3
48	Tareq	22	67	3
54	Ahmad	21	81	2

جدول المواد course

cid	cdesc	credit	prerequisite
10	english 1	3	0
11	English 2	2	10
101	calculus 1	3	0
102	calculus 2	3	101
201	programming 1	4	0
202	programming 2	4	201
301	data structure	3	202

الطلاب درس المادة- جدول enroll

cid	sid	mark
10	21	70
11	21	69
10	27	78
11	27	75
101	27	79
102	27	60
101	29	70
201	29	77

```
SELECT sname FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND ( C.cdesc ='calculus 1' OR C.cdesc ='calculus 2')
```

sname
Mohammed
Mohammed
Tawfiq

# دمج البيانات بين استعلامين- الاتحاد UNION

```
SELECT sname FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND ( C.cdesc ='calculus 1' OR C.cdesc ='calculus 2')
```



يمكن استخدام عملية الاتحاد UNION كالآتي:

```
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc ='calculus 1'
UNION
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc ='calculus 2'
```

جملة الاستفسار الأولى تعطينا المجموعة {محمد ،  
توفيق} اما جملة الاستفسار الثانية تعطينا {محمد}

الاتحاد بين المجموعتين يعطينا {محمد ، توفيق}  
كما هو مبين في الجدول أعلاه، نلاحظ هنا انه  
عند استخدام العمليات على المجموعات فان  
النتيجة تكون بدون تكرار.

sname
Mohammed
Tawfiq

# دمج البيانات بين استعلامين- التقاطع INTERSECTION

التعبير Intersect:

يُستخدم التعبير Intersect لاستعادة السجلات التي تظهر في كل من السجلات التي يعيدها الاستعلام الأول والسجلات التي يعيدها من الاستعلام الثاني.

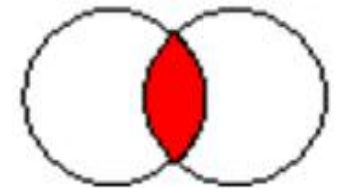
يُستخدم التعبير Intersect وفق الصيغة:

```
select columnA,columnB from tableA  
Intersect  
Select columnC,columnD from tableB;
```

# دمج البيانات بين استعلامين- التقاطع INTERSECTION

يمكننا ان نعيد صياغه المثال السابق باستخدام التقاطع: المطلوب عرض أسماء الطلاب اللذين أنهوا دراسة المادتين 'calculus 1' و 'calculus 2'

```
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND ( C.cdesc ='calculus 1' AND C.cdesc ='calculus 2')
```



يمكن استخدام عملية التقاطع INTERSECTION كالآتي:

```
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc ='calculus 1'
INTERSECT
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc ='calculus 2'
```

التقاطع بين المجموعتين {محمد ، توفيق} ∩ {محمد} = {محمد}

sname
Mohammed

# دمج البيانات بين استعلامين- الفرق EXCEPT

التعبير Except:

يُستخدم التعبير Except لإعادة السجلات التي تظهر في نتيجة الاستعلام الأول ولا تظهر في نتيجة الاستعلام الثاني.

يستخدم التعبير Except وفق الصيغة:

```
select columnA,columnB from tableA  
Except  
Select columnC,columnD from tableB;
```

# دمج البيانات بين استعلامين- الفرق EXCEPT

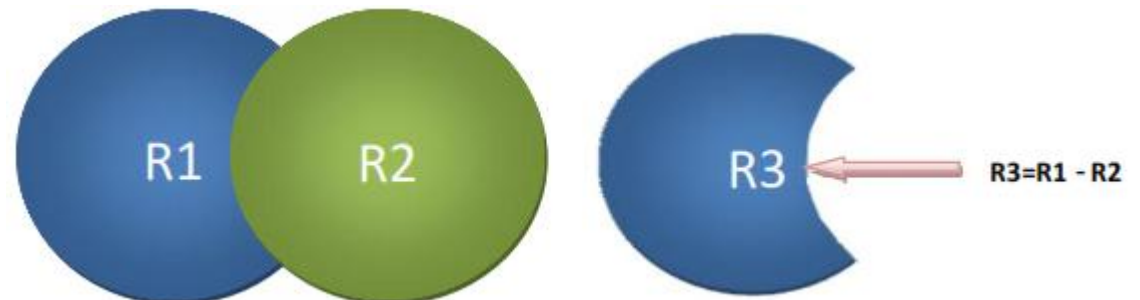
مثال: المطلوب عرض اسماء الطلاب اللذين درسوا المادة 'calculus1' ولم يدرسوا 'calculus2'

يمكن استخدام عملية ماعدا EXCEPT والتي تقابل difference كالآتي:

```
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc = 'calculus 1'
EXCEPT
SELECT sname
FROM student S, course C, enroll E
WHERE S.sid = E.sid
AND E.cid = C.cid
AND C.cdesc = 'calculus 2'
```

الاستفسار الأول يعطينا المجموعة {محمد ، توفيق} أما  
الاستفسار الثاني {محمد} وبالتالي المجموعة الاولى -  
المجموع الثانية هي المجموعة {توفيق}

sname
Tawfiq



# أنواع الربط join. 1- الربط الطبيعي Natural join أو inner join

مثال : باستخدام الربط - المطلوب عرض كل الحقول للموظفين التابعين لأقسام معينة.

Table "employee"		Table "department"	
LastName	DepartmentID	DepartmentName	DepartmentID
Smith	34	Sales	31
Jones	33	Engineering	33
Robinson	34	Clerical	34
Jasper	36	Marketing	35
Steinberg	33		
Rafferty	31		

```
SELECT * FROM employee [INNER] JOIN department
      ON employee.DepartmentID = department.DepartmentID
```

تكافئ التالي باستخدام WHERE

```
SELECT * FROM employee, department
      Where employee.DepartmentID = department.DepartmentID
```

LastName	DepartmentID	DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Steinberg	33	Engineering
Rafferty	31	Sales



## الربط الخارجي Outer Joins

- لأخذ جميع السجلات من الجدول الأول Table1 وفقط السجلات من الجدول الثاني Table2 التي تتطابق فيها قيمة الحقل Column1 من الجدول Table1 مع قيمة الحقل Column2 من الجدول الثاني Table2، نكتب الصيغة:

```
Select * from Table1 LEFT OUTER JOIN Table2  
ON Table1.Column1 = Table2.Column2;
```

- لأخذ جميع السجلات من الجدول الثاني Table2 وفقط السجلات من الجدول الأول Table1 التي تتطابق فيها قيمة الحقل Column1 من الجدول Table1 مع قيمة الحقل Column2 من الجدول الثاني Table2، نكتب الصيغة:

```
Select * from Table1 RIGHT OUTER JOIN Table2  
ON Table1.Column1 = Table2.Column2;
```

- لأخذ جميع السجلات من الجدول الثاني Table2 وجميع السجلات من الجدول الأول Table1 بحيث نتوضع السجلات التي تتطابق فيها قيمة الحقل Column1 من الجدول Table1 مع قيمة الحقل Column2 من الجدول الثاني Table2 في نفس السجل من جدول القيم المعادة، نكتب الصيغة:

```
Select * from Table1 FULL OUTER JOIN Table2  
ON Table1.Column1 = Table2.Column2;
```

# الربط الخارجي Outer Joins

## الربط الخارجي Outer Joins

الربط العادي حينما يكون هناك سجل ليس له اي مقابل حسب شرط الربط في الجدول الاخر لا يظهر السجل، لغة SQL تعطينا امكانية اخرى باستخدام القيم الخالية وهي ان يظهر كل سجل في الجدول الاول مرة على الاقل حتى ولو لم يكن له مقابل في الجدول الاخر مع وجود قيم خالية مكان الحقول القادمة من الجدول الثاني التي ستظهر في النتيجة.

مثال :

Table "employee"		Table "department"	
LastName	DepartmentID	DepartmentName	DepartmentID
Smith	34	Sales	31
Jones	33	Engineering	33
Robinson	34	Clerical	34
Jasper	36	Marketing	35
Steinberg	33		
Rafferty	31		

## الربط الخارجي Left outer join

مثال : المطلوب عرض كل الحقول للموظفين التابعين والغير تابعين للاقسام.

Table "employee"		Table "department"	
LastName	DepartmentID	DepartmentName	DepartmentID
Smith	34	Sales	31
Jones	33	Engineering	33
Robinson	34	Clerical	34
Jasper	36	Marketing	35
Steinberg	33		
Rafferty	31		

```
SELECT * FROM employee LEFT OUTER JOIN department
ON employee.DepartmentID = department.DepartmentID
```

LastName	DepartmentID	DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Jasper	36	<b>NULL</b>
Steinberg	33	Engineering
Rafferty	31	Sales

## الربط الخارجي Right outer join

مثال : المطلوب عرض كل الحقول للاقسام التي يتبعها موظفين والتي لا يتبعها موظفين.

Table "employee"		Table "department"	
LastName	DepartmentID	DepartmentName	DepartmentID
Smith	34	Sales	31
Jones	33	Engineering	33
Robinson	34	Clerical	34
Jasper	36	Marketing	35
Steinberg	33		
Rafferty	31		

```
SELECT * FROM employee Right OUTER JOIN department
ON employee.DepartmentID = department.DepartmentID
```

LastName	DepartmentID	DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Steinberg	33	Engineering
Rafferty	31	Sales
<b>NULL</b>	35	Marketing

## الربط الخارجي Full outer join

مثال : المطلوب عرض كل الحقول للموظفين التابعين والغير تابعين لاقسام معينة وكذلك للاقسام التي يتبعها موظفين والتي لا يتبعها موظفين.

Table "employee"	
LastName	DepartmentID
Smith	34
Jones	33
Robinson	34
Jasper	36
Steinberg	33
Rafferty	31

Table "department"	
DepartmentName	DepartmentID
Sales	31
Engineering	33
Clerical	34
Marketing	35

```
SELECT * FROM employee Full  
OUTER JOIN department  
ON employee.DepartmentID =  
department.DepartmentID
```

LastName	DepartmentID	DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Jasper	36	<b>NULL</b>
Steinberg	33	Engineering
Rafferty	31	Sales
<b>NULL</b>	35	Marketing

# تمارين في الجبر العلائقي و لغة SQL

## • الجداول

- Professor(ssn, profname, status)
- Course(crscode, crsname, credits)
- Taught(crscode, semester, ssn)

- الاستاذ ( الرقم، الإسم، الحالة)
- المواد (رقم المادة، إسم المادة، الوحدات)
- يعلم (رقم المادة، الفصل الدراسي، رقم الاستاذ)

# تمارين في الجبر العلائقي و لغة SQL

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

• الجداول

## الاستفسار Query 1

• المطلوب أرقام الاساتذة الذين لم يدرسوا المقرر **csc7710**.

• الحل بالجبر العلائقي

$\pi_{ssn}(\text{Professor}) - \pi_{ssn}(\sigma_{crscode='csc7710'}(\text{Taught}))$

• الحل SQL

(**SELECT** ssn **From** Professor) **EXCEPT** (**SELECT** ssn **From** Taught T  
**Where** T.crscode = 'CSC7710')

# تمارين في الجبر العلائقي و لغة SQL

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

• الجداول

## الاستفسار 2 Query

• المطلوب أرقام المواد الدراسية التي لم تدرس.

• الحل بالجبر العلائقي

$\pi_{\text{crscode}}(\text{Course}) - \pi_{\text{crscode}}(\text{Taught})$

• الحل SQL

(**SELECT** crscode **FROM** Course) **EXCEPT** (**SELECT** crscode **FROM** TAUGHT)



# تمارين في الجبر العلائقي و لغة SQL

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

• الجداول

## الاستفسار 3 Query

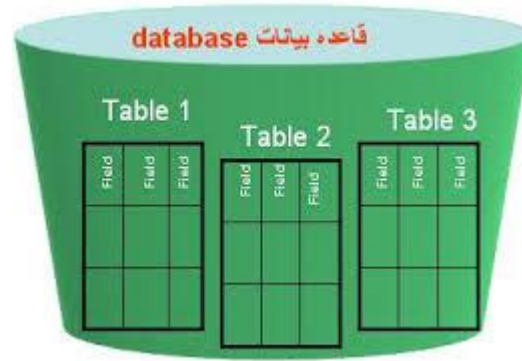
• المطلوب أسماء الاساتذة الذين درسوا المادة **CSC6710**.

• الحل بالجبر العلائقي

$\pi_{\text{profname}}(\sigma_{\text{crscode}='csc6710'}(\text{Taught}) \bowtie \text{Professor})$

• الحل SQL

**SELECT** P.profname **FROM** Professor P, Taught T  
**WHERE** P.ssn = T.ssn **AND** T.crscode = 'CSC6710'



نهاية المحاضرة

Any Questions